THE UNIVERSITY OF MICHIGAN

Technical Report 24

CONCOMP

July 1970

SELCIR: SYSTEMS ENGINEERING LABORATORY CIRCUIT-DRAWING PROGRAM

James F. Blinn

THE UNIVERSITY OF MICHIGAN

Technical Report 24

SELCIR: SYSTEMS ENGINEERING LABORATORY
CIRCUIT-DRAWING PROGRAM

James F. Blinn

CONCOMP: Research in Conversational Use of Computers
F. H. Westervelt, Project Director
ORA Project 07449

supported by:

ADVANCED RESEARCH PROJECTS AGENCY
DEPARTMENT OF DEFENSE
WASHINGTON, D.C.

CONTRACT NO. DA-49-083 OSA-3050
ARPA ORDER NO. 716

administered through:

OFFICE OF RESEARCH ADMINISTRATION   ANN ARBOR

July 1970

TABLE OF CONTENTS

ABSTRACT

SELCIR, Systems Engineering Laboratory Circuit Drawer,
enables the user to draw electronic circuit diagrams on a CRT display.
All graphical manipulations are performed locally in a PDP-9, which,
together with a 339 display scope, forms a remote terminal to an
IBM 360/67. SELCIR scans the drawn network and transmits the
topology of the network diagram to the central computer. The 360
then analyzes the network and sends back results which are displayed
in the form of graphs. To enable the user to express himself naturally,
the graphical manipulations emphasize the use of the light pen rather
than push buttons. Also the user can stop at any point to correct
mistakes or make changes. Finally, many constraints on the drawing
process enable him to quickly make a neat looking diagram.

1. INTRODUCTION

This report describes the operation of SELCIR, first from the user's point of view (an operator's manual), and second from the programmer's point of view (a programming manual).

Section 2 (User's Manual) describes the basic operation of the hardware and the user's interaction with the program. Details of the constraints placed on the user are kept to a minimum as they only keep him from doing things he would not really want to do. The procedure for transmitting the network to the 360 and for specifying the output format are described in this section.

Section 3 (Internal Operation) describes the programming techniques used which may be usefully applied to other programs. Included is information concerning the internal structure of the program, its tasks, data structure, thresholding, etc. Reference [1] (which describes the PDP9 executive system under which SELCIR runs) is helpful but not essential to understanding this section.

1

2. USER'S MANUAL

2.1 Background Material

2.1.1 Light Pen

Most of the communication between the user and SELCIR is made with the light pen, a device having a photocell behind a shutter. When the light pen is pointed at the screen and the shutter is opened (by pushing the button on the side) the photocell can sense light from the screen, which is then interpreted by the PDP9. Thus, at any time, the light pen is enabled to recognize certain parts of the displayed picture, and the program can perform functions on it.

The light pen generally will specify either "what" or "where." By pointing at an object on the screen and opening the shutter, the user can specify "what" object to operate on. At times when the light pen is being asked "where", a small cross is displayed under the light pen. As long as the pen sees the cross, the program will continually adjust its position to be directly under the light pen. Thus, the cross will follow the light pen around on the screen in a process called "tracking." If the shutter of the light pen closes while tracking is in progress, the cross can no longer be seen. A brief flash appears as a search pattern is displayed in a desperate attempt to find the light pen. This is termed "loss of tracking" and signals the program to terminate an operation.

2.1.2 Light Buttons and Light Switches

Throughout SELCIR's operation, a variety of words appear on the screen. Pointing the light pen at a word will initiate a particular action, generally described by what the word says. These words are called "light buttons."

In several instances, a light button indicates one of two possible alternatives. Pointing to the light button flips it to the other choice available. Pointing again flips it back to the original choice. This arrangement is called a "light switch" (since it operates like a toggle switch) with the light buttons as labels for the two positions. At all times, the light switch displays the currently chosen alternative and further action in the program may be based on this choice.

2.1.3 Getting SELCIR on the Air

To start the program, the following steps must be taken (only underlined characters should be typed on the teletype):

1. Ensure that the SEL EXECUTIVE SYSTEM is running and in system state (when in doubt, hit the INTERRUPT button next to the teletype).

2. Place the SELCIR tape into the high speed paper tape reader.

3. Type FROM PAPER TAPE TO CORE command.

4. Type SCHEDULE 20000 command.

5. Type RUN command.

The screen will then look like this:
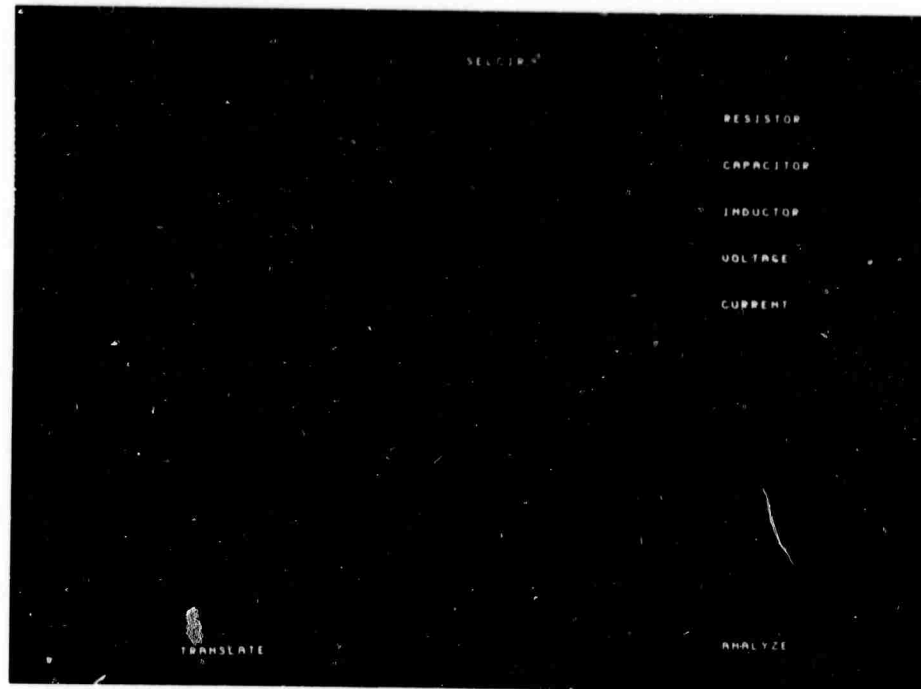
2.2 Creating Circuit Elements

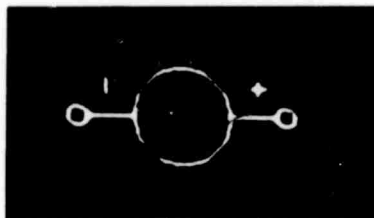By pointing to one of the right-hand light buttons, the elements of the circuit can now be created. The created circuit element will then follow the motion of the light pen along with the tracking cross until tracking is lost. The elements, as they are created, appear as:
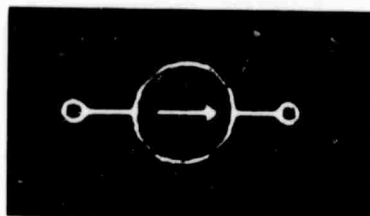
Parameter dot

terminal → ← terminal

body of
element

RESISTOR
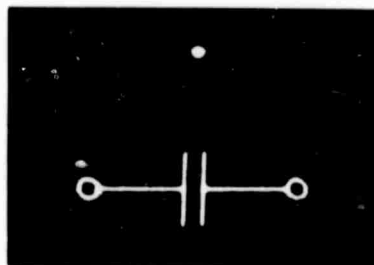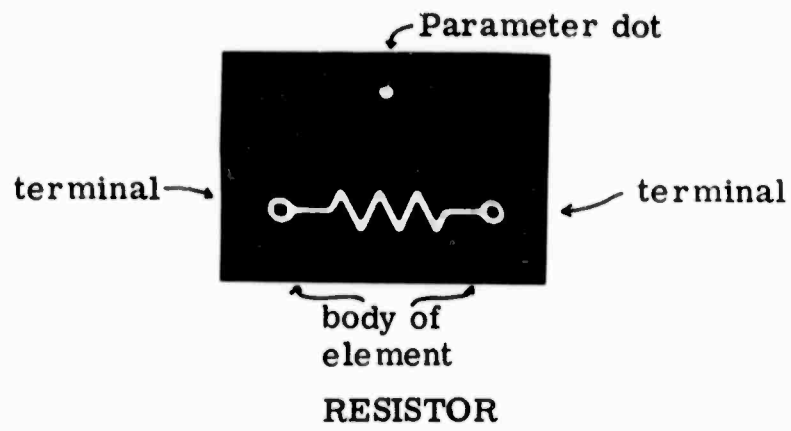
CAPACITOR
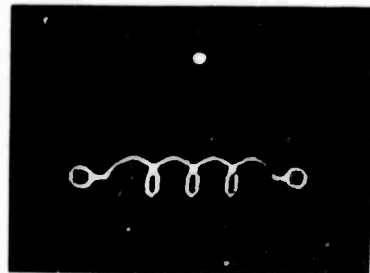
INDUCTOR

VOLTAGE SOURCE

CURRENT SOURCE

Available Circuit Elements

2.3 Operations on Existing Elements

An element on the screen can be moved, deleted, assigned a parameter value, rotated, or connected to other elements by following the procedures outlined below.

2.3.1 Move

An element is moved by pointing to the body of the element and moving the pen about an inch-and-a-half away from it. They effectively "yanks" the element away from its current position. It will then follow the light pen, as it did upon creation, until tracking is lost.

2.3.2 Delete

An element may be deleted by pointing to it and stroking up and down with a "rub-out" motion. After five strokes, the element disappears.

2.3.3 Rotate

An element is rotated by pointing to it and making a circular motion around it. When a complete circle is made, the element will rotate in 90° increments to follow the circular motion of the light pen. The difference, then, between moving, deleting, and rotating, lies in the motion of the light pen after pointing to the element. Whichever distinctive motion is detected first determines the operation to be performed.

Move/Delete/Rotate

Element on screen

point to it with light pen
(tracking cross appears)
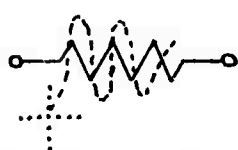
Stroke up and down
five times
    or
Make circular
motion
    or
Move light pen
away from element

Element disappears

Element rotates

Element moves to tracking cross
and follows it from then on

2.3.4 Assign Value

The passive elements (resistor, capacitor, and inductor) can have numeric values assigned to them (resistance, capacitance, or inductance). The number is specified by entering its digits on the push-button box. The digits are "typed" with:
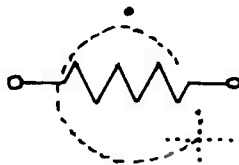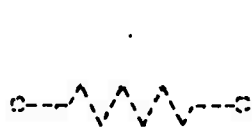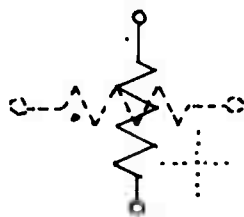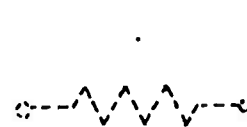
buttons 0 - 9   meaning digits 0 - 9

button 10   meaning decimal point

button 11   meaning delete currently typed parameter

As the number is typed in, it appears in the upper right-hand corner of the screen. All parameters must have a decimal point. When the desired parameter is entered, the user specifies which element is to receive it by pointing with the light pen to the parameter dot or to an already assigned parameter of an element. The number in the upper right-hand corner of the screen then appears as the new parameter of the element.

2.3.5 Assign Control

Voltage and current sources (as they are created) are independent sources. They can be changed to controlled sources by pointing and immediately dropping tracking (before a specific motion is made to move, delete, or rotate). A light switch will appear at the upper left corner of the screen to allow the choice between control by VOLTAGE ACROSS or by CURRENT THROUGH the control element. The user then specifies the desired control element by pointing to it with the

light pen, and a set of labels appears near the affected elements to indicate their relationship. The control is by voltage across or current through the controlling element depending on the state of the light switch when the controlling element was chosen. The light pen then returns to move/delete/rotate mode. A source can be changed from controlled back to independent by specifying it as its own control.

2.3.6 Connect

Connections are made between elements by drawing lines between them. A connection is begun by pointing to a terminal on the end of an element. The connection line will replace the terminal, and its endpoint will follow the light pen. The lines are constrained to be always horizontal or vertical. If the pen is moved horizontally the line is drawn horizontally; if the pen is moved vertically, the line is drawn vertically. When the pen "moves around a corner" the connection line is broken and a new segment added.

Element
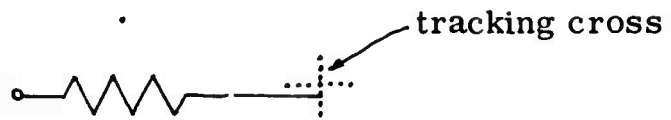
Beginning connection

tracking cross
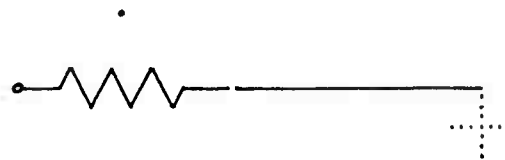
Move horizontally
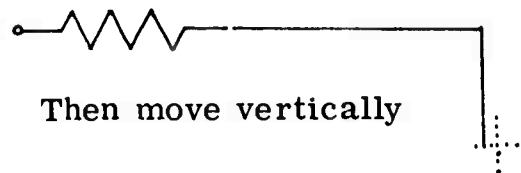
Then move vertically

Moving the pen backward over a line "swallows it up. " Thus,
segments can be deleted by reversing the pen motion.

Four line segments

Shrinking a line

Deleting a segment

Deleting a segment

Adding a new segment
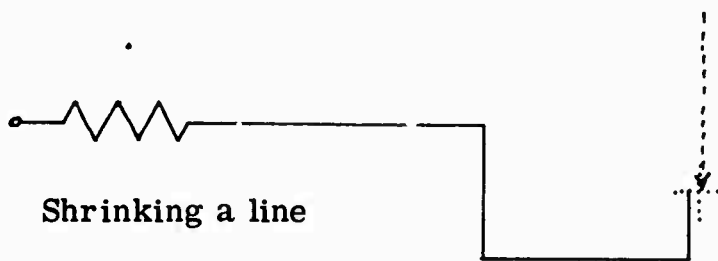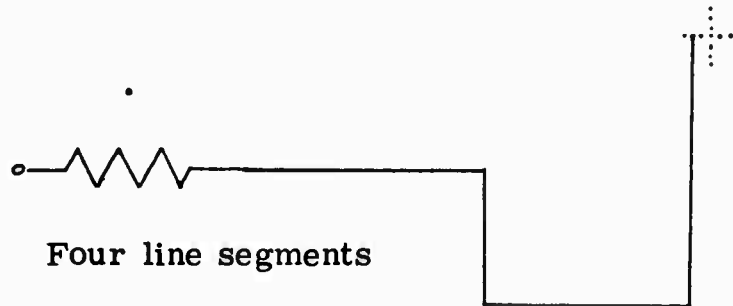
The connection line continues to be drawn until the light pen runs into another terminal, whereupon the endpoint of the line and the position of the connected elements are adjusted to match. The elements are then topologically and pictorially connected. Moving a connected element moves all connection lines and elements connected to it. Deleting a connected element deletes the connection lines that terminate at that element but does not delete the other elements connected to it. Connected elements cannot be rotated.
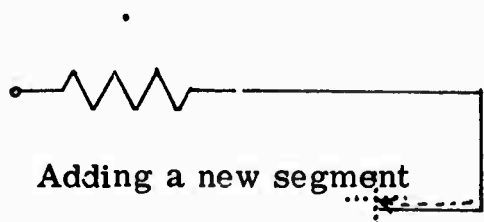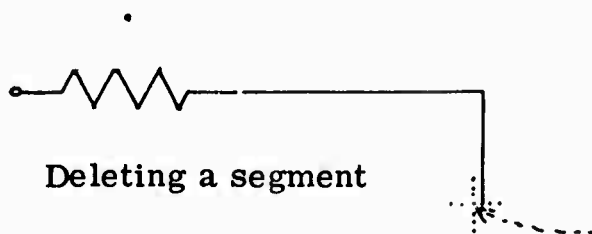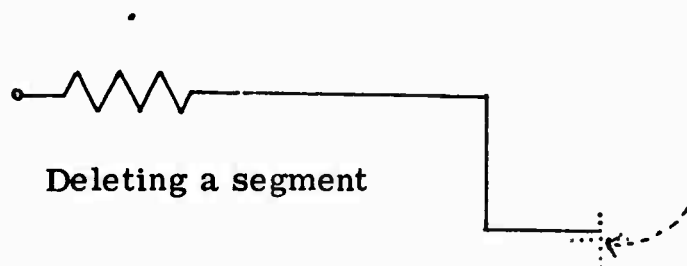
If tracking is lost while a line is being drawn or if the user attempts to draw a line in an obviously ridiculous direction, the connection line will disappear and the connection process is discontinued. The light pen is immediately re-enabled to interpret the diagram as usual.

Pointing to a completed connection line causes a junction dot to appear on the line. A flash of light will appear at the point where the junction dot was put (to call attention to it). Then a procedure ensues similar to that of move/delete for elements. If five strokes are made up and down across the dot, it and all connection lines radiating from it are deleted. If the pen is moved an inch and a half away from the junction dot, a branch line will be drawn from the junction. This line follows the light pen and connects to elements in the same fashion as one starting from a terminal. It will continue to be drawn until tracking is lost (line deleted) or a terminal is encountered (connection is completed). Note that here, too, the decision to branch or delete

is made by whichever motion happens first. Thus shaking a branch line up and down over a junction will not delete the junction.

Connection line
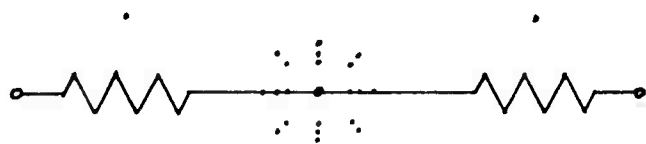
Creation of junction

OR

Scratch up and down
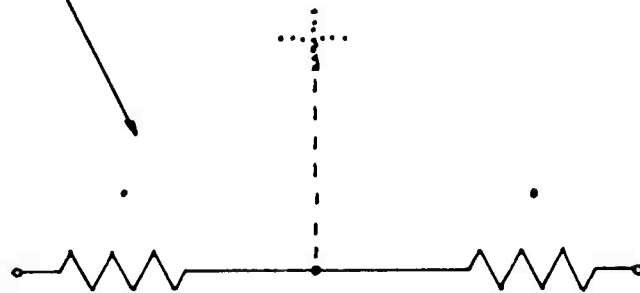
Move away

Deleting connection

Creating branch

Note that in deleting connection lines, rubbing out deletes only the branch of the connection which is pointed to, as shown below.

Connection line

Point to connection line and rub out

Only one line disappears

OR

Point to junction already in existence and rub out
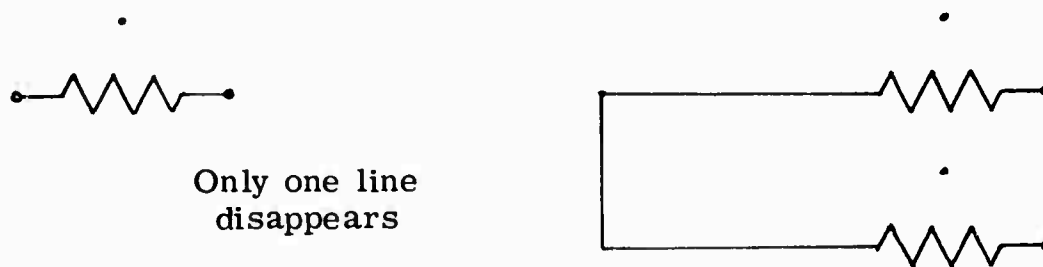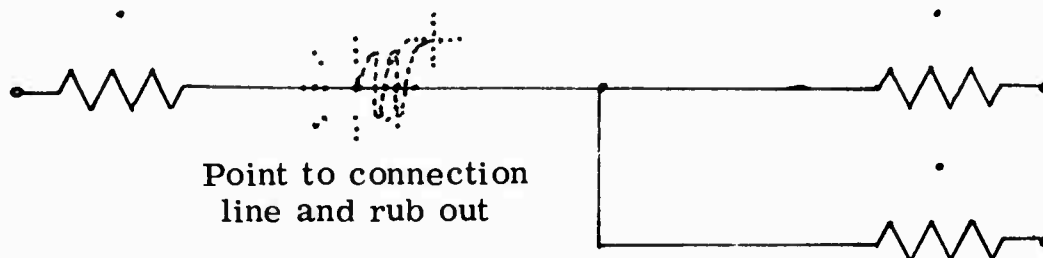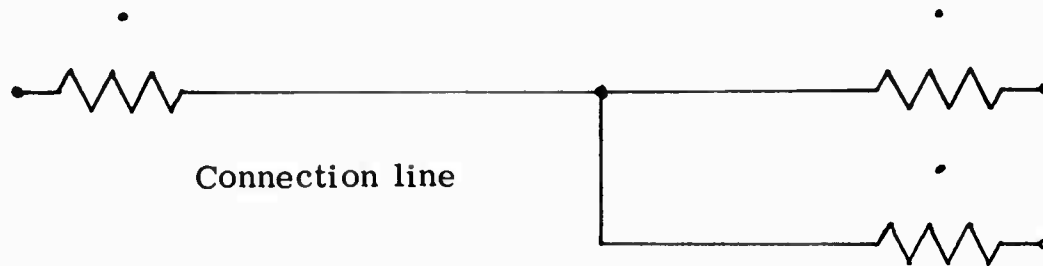
Junction and all lines radiating from it disappear
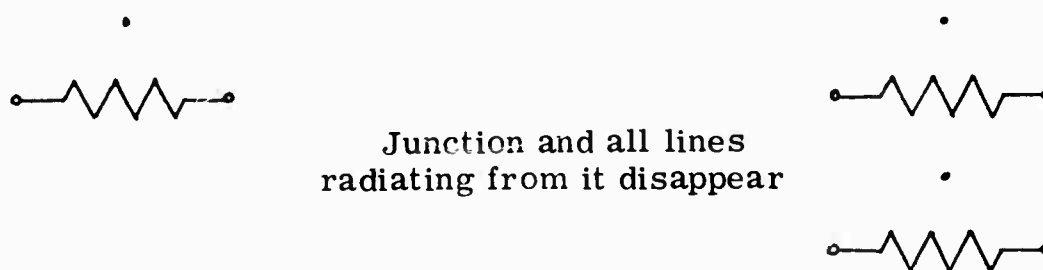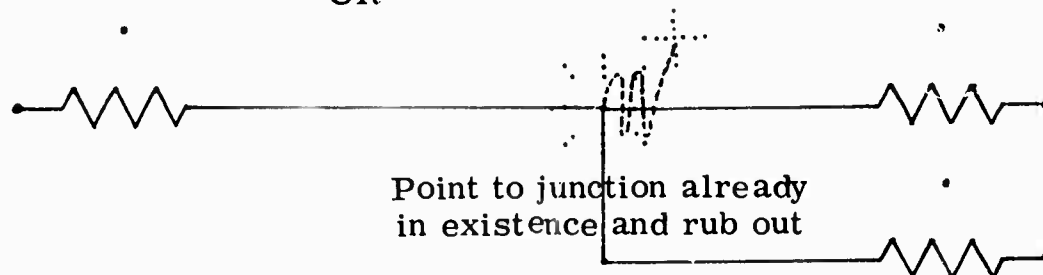
There is another way of terminating a connection. While a line is being drawn, the light pen is looking at other terminals and other connections. If it sees another connection line, a junction is created at that point. The light pen is still enabled to see terminals, and will connect to them if it does, but if tracking is lost, the connection being drawn will connect into the junction dot just created. If, without losing tracking, the connection line is drawn on past the newly created junction, the program decides that a junction was not called for and it is removed. This facility allows connections to cross over other connections as well as to be joined to them.

2.4 Abnormal Conditions during Diagram Drawing

Because of limited core storage in the PDP9, the user may attempt to draw a diagram larger than can fit in memory. (Currently about 13 elements can be created and connected.) In this case, the comment "NO ROOM" is typed on the teletype and the current operation is aborted.

2.5 Summary of Diagram-drawing Operations

CREATE ELEMENT	-	Point to element name, Move to desired location Drop tracking
DESTROY ELEMENT	-	Point to element Stroke up and down five times
MOVE ELEMENT	-	Point to element Move 1-1/2 inches away Move to desired location Drop tracking

16

Connection being drawn
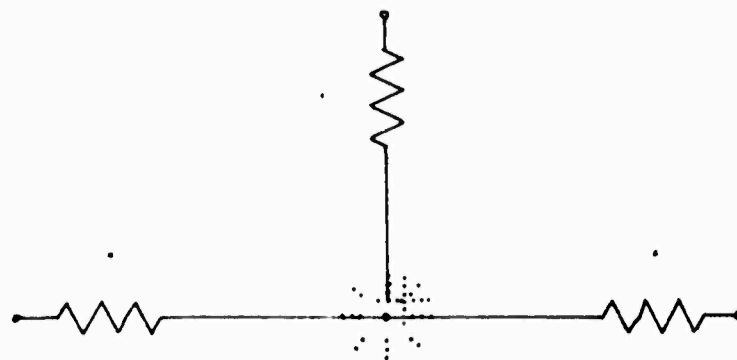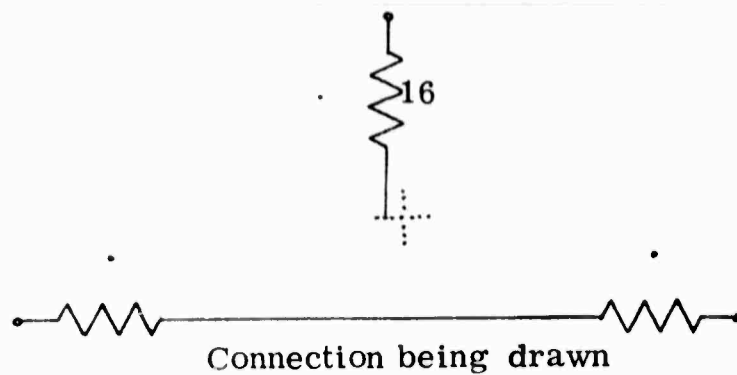
Break in other connection

or

dot disappears
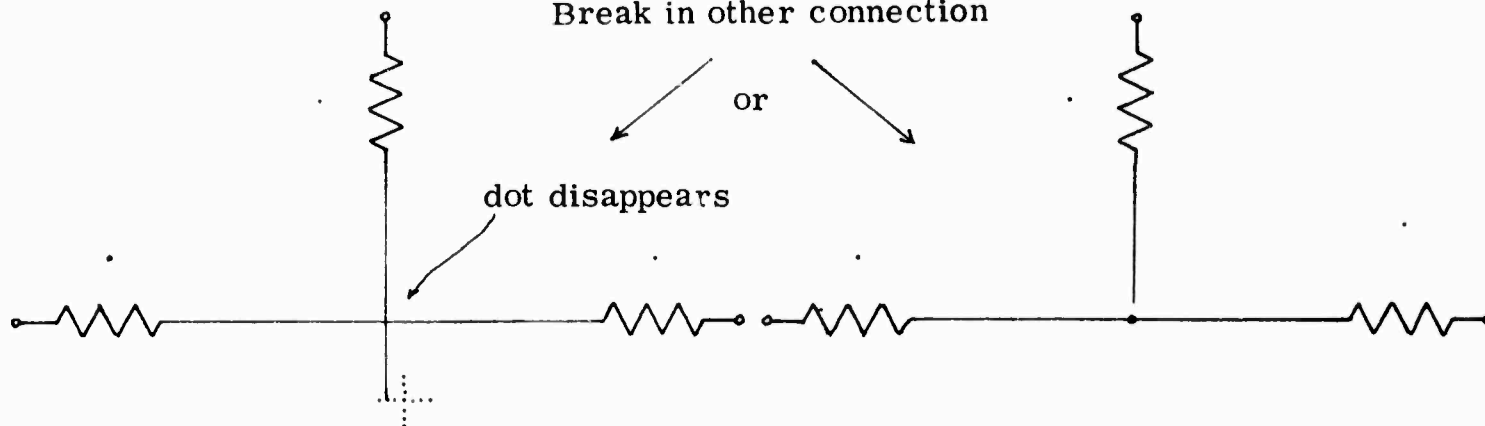
Moved away from dot

Loss of tracking
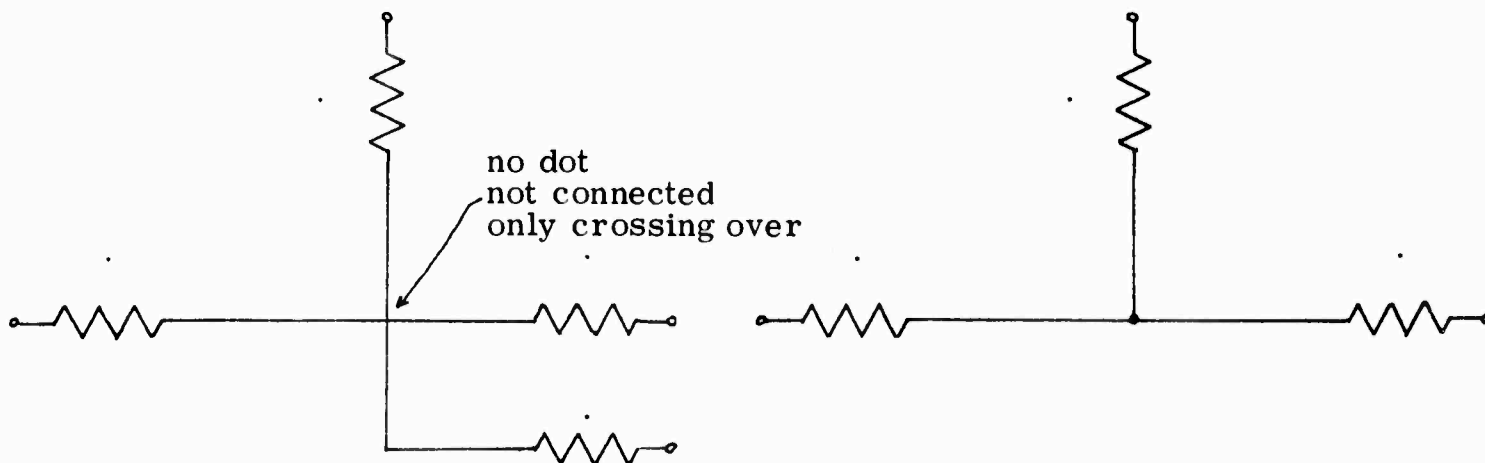
no dot
not connected
only crossing over

Connected to something else

Connection made

ROTATE ELEMENT	-	Point to element Make circular motion around it
CHANGE } ASSIGN } PARAMTER	-	Type value on pushbottons (with ". ") Point to element parameter or parameter dot
BEGIN CONNECTION	-	Point to terminal of element

<div align="center">OR</div>

Point to already drawn connection
and move 1-1/2 inches away

COMPLETE CONNECTION	-	Draw line to another element terminal

<div align="center">OR</div>

Draw to an existing connection
line
Lose tracking

DELETE CONNECTION LINE	-	Point to line or junction Stroke up and down five times
CREATE CONTROLLED SOURCE AND ASSIGN CONTROL ELEMENT	-	Create independent source Point to it and drop tracking immediately Flip VOLTAGE ACROSS vs. CURRENT THROUGH light switch to desired state Point to control element

2.6 Diagram Translation

In making his drawing, the user is not limited to the actual screen size; the available size is actually eight screen widths by eight screen widths. Any elements or circuit fragments moved off the edge of the screen with the light pen will appear chopped off at the edge; they still exist, however, and are transmitted to the 360 and enter into the

analysis.   They again become visible when moved back on the screen,
as shown below.
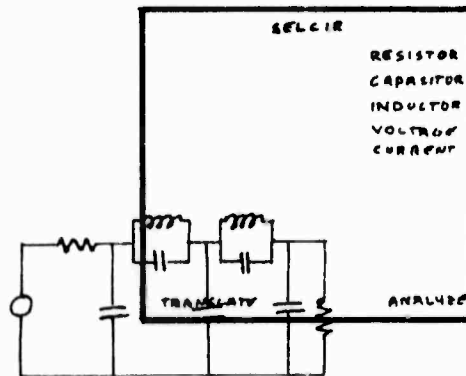
The area inside the heavy box fills the screen and the rest is off
the edge.

The TRANSLATE light switch accompanies the above-mentioned
translation feature.   Normally, unconnected circuit elements are
moved independently by the light pen.   Pointing to TRANSLATE flips
it to TRANSLATE ON and all circuit elements, interconnected or not,
move when the light pen points to any one of them.   Pointing to
TRANSLATE ON flips it to TRANSLATE and the light pen operates in
the normal move/delete/rotate mode.   This feature is especially
useful if an element is "dropped" on top of a creation light button.
Pointing to the element to move it usually is interpreted as pointing,
instead, to the creation light button.   To move such an element away
from the creation light buttons, the user merely flips TRANSLATE to

TRANSLATE ON, points to an out-of-the-way element and moves it, thereby moving all circuit elements on the screen. Thus, the rest of the diagram can be used as a "handle" to move the offending element away from the creation light buttons.

2.7 Teletype Operation

Meanwhile, at the teletype other operations can be performed. The Clear and Escape operations are initiated from the teletype because they are rather drastic and require the user to think twice before doing them. Only the underlined characters should be typed on the keyboard; the program will type the rest. All invalid characters typed for keyboard commands are totally ignored.

CLEAR?	OK	Erase the network diagram currently on the screen and clear out the internal name table.
CLEAR?	NO	Chicken out from CLEAR? Does nothing
ESCAPE?	OK	Terminates execution of SELCIR Sends End of File to 360
ESCAPE?	NO	Chicken out from ESCAPE? Does nothing
# ...		Copy rest of line to 360

In the last case, the typed line is transmitted over the dataphone when a carriage return is hit. This facility is mainly for sending 360 control commands while SELCIR is running. Special line-editing characters interpreted while the line is being typed are:

"←"   delete previous character (i.e., backspace)

RUB-OUT   delete whole line

RETURN   End of line, transmit it to 360

The character set which can be typed at the keyboard and interpreted by SELCIR is limited by the executive system (See [1]). Therefore, a special operation is taken to allow certain special characters to be typed. The character "↑" (suggesting upper shift) is taken as a "non-literal next" character. The character immediately following it is given a special interpretation:

Characters typed	As seen by 360
↑N	↑
↑P	@
↑A	Control A
↑B	Control B
↑E	Control E (Attention)
↑Y	Control Y (End of File)

2.8 Diagram Output and Results Specification

When the diagram is completely drawn, the user specifies the desired output format of the results of the analysis. He switches from construct mode to analyze mode by pointing to ANALYZE at the lower right-hand corner of the screen. The screen will then look like this:
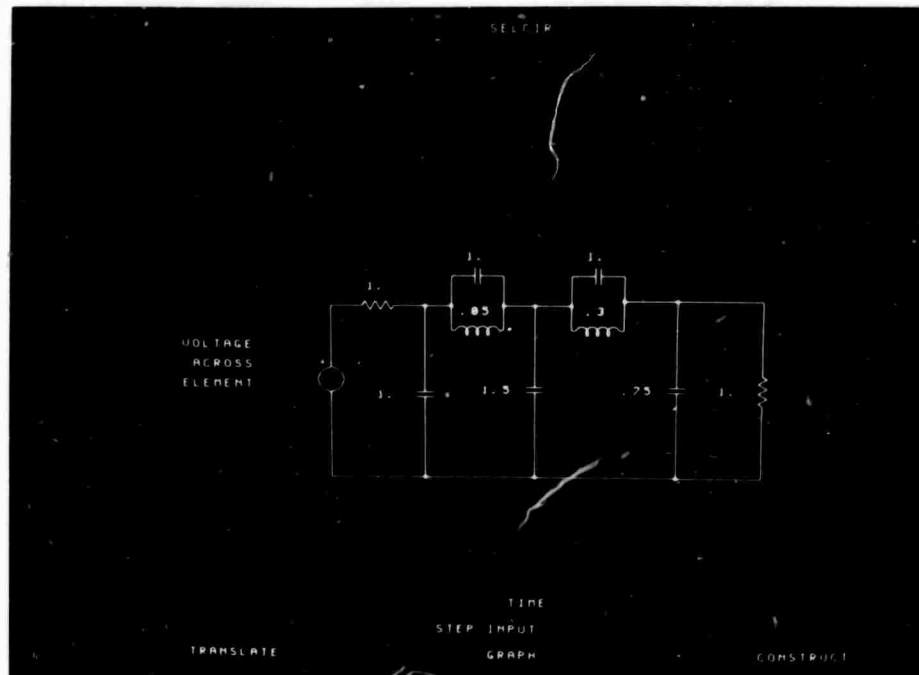
A modified version of CIRAN, an electronic circuit analysis program
in use at the University of Michigan, produces graphs of various
properties of the circuit in response to the circuit description
generated by SELCIR. CIRAN runs on an IBM 360/67 under a time-
sharing system called MTS (Michigan Terminal System) [2]. Thus
the user must first issue the appropriate commands from the teletype
(prefixed by a #) to cause MTS to read the diagram description from
SOURCE. For example:

$COPY   *SOURCE* ⭠ PSP TO -T

copies the diagram description to the file "-T" for later use.

$RUN CIRAN 5 = *SOURCE*↑PSP

runs the analysis program which will read the diagram description
as data from the PDP-9. Note that whenever *SOURCE* is used in
this way, the "↑PSP" (seen by MTS as @SP) device modifier should
be used. This disables the sending of a prompting character (a blank)
to the PDP-9 every time a line is read. Such a prompting character
could cause line contention problems on the dataset and freeze up
both programs.

The user must then specify the circuit parameters he wants
graphed. While SELCIR is in analyze mode, there is an x-axis label
along the bottom edge of the screen and a y-axis label along the left
edge. These labels are actually light switches whose states indicate
the current results request. The independent variable to be plotted
is determined by the state of the x-axis label, a light switch at the
lower center of the screen. Pointing to it will sequence through:
TIME, FREQUENCY, POLES AND ZEROS, TIME ...

TIME - The maximum and minimum times to be plotted are
specified by parameter dots to the right and left of TIME. The
values are entered in the same way as specifying element parameter
values. CIRAN assumes the existence of only one independent source
which can be given a STEP INPUT or an IMPULSE INPUT (chosen by
the light switch which appears below TIME) at time $t = 0$. The y axis
will be a plot of VOLTAGE ACROSS or CURRENT THROUGH (chosen

by a light switch at the left side of the screen) one of the circuit elements.

FREQUENCY - The maximum and minimum frequencies plotted are specified as above. The frequency can be plotted on a LOGA-RITHMIC SCALE or a LINEAR SCALE (light switch which appears below FREQUENCY). The y axis will be a plot of GAIN (or PHASE) of the VOLTAGE ACROSS (or CURRENT THROUGH) the element. The gain can be plotted as DB or ABSOLUTE gain (all chosen by light switches which appear at the left side of the screen.)

POLES AND ZEROS - This is a plot of the poles and zeros of the complex s-plane used on the Laplace transform analysis of the circuit. It is generated for the VOLTAGE ACROSS or CURRENT THROUGH an element (light switches at the left side of the screen).

Various interlocks within the program guarantee that the proper set of light switches are on the screen to match the x-axis label. Thus, it is impossible to request a STEP IMPULSE on a FREQUENCY plot, etc. The possible valid choices of graphs are:

$$
\text{TIME} \begin{Bmatrix} \text{STEP INPUT} \\ \text{IMPULSE INPUT} \end{Bmatrix} \text{VS.} \begin{Bmatrix} \text{VOLTAGE ACROSS} \\ \text{CURRENT THROUGH} \end{Bmatrix} \text{ELEMENT}
$$

FREQUENCY $\begin{Bmatrix} \text{LINEAR} \\ \text{LOGARITHMIC} \end{Bmatrix}$ SCALE VS. $\begin{Bmatrix} \begin{Bmatrix} \text{ABSOLUTE} \\ \text{DB} \end{Bmatrix} \begin{matrix} \text{PHASE} \\ \text{GAIN} \end{matrix} \end{Bmatrix}$ OF $\begin{Bmatrix} \text{VOLTAGE ACROSS} \\ \text{CURRENT THROUGH} \end{Bmatrix}$ ELT

POLES AND ZEROS OF S-PLANE FOR $\begin{Bmatrix} \text{VOLTAGE ACROSS} \\ \text{CURRENT THROUGH} \end{Bmatrix}$ ELEMENT

For example, the specification of a graph of the gain (in db) of the current through an element plotted against frequency on a linear scale from .01 to 1.0 would look like this:
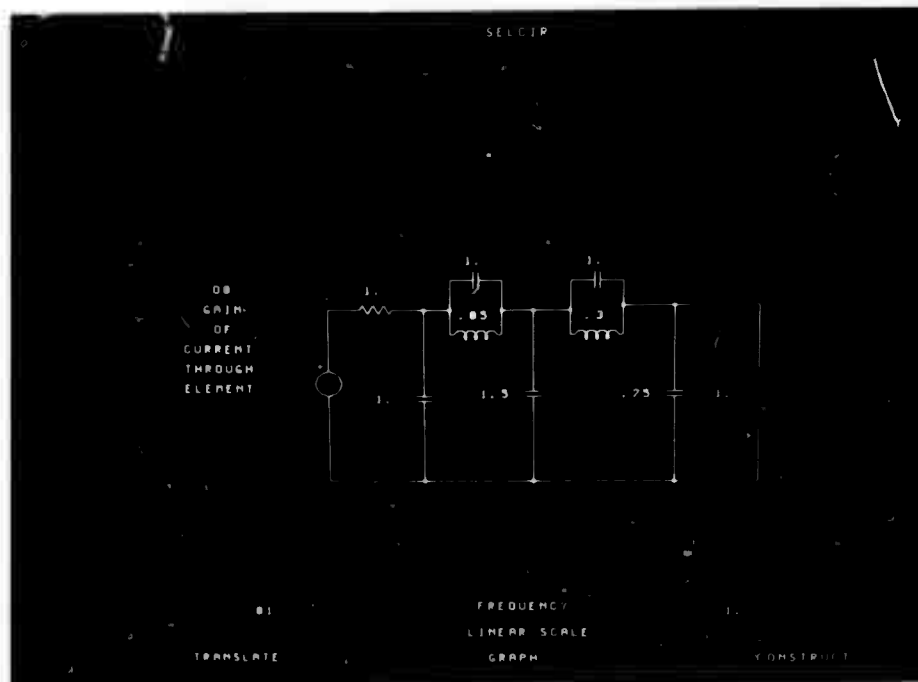
The user selects the element he wishes plotted by pointing to it, causing it to blink. The output element can be changed by pointing to a new selection.

Then, pointing to GRAPH causes a set of error checks to be made (see Section 2.9) and, if no errors are detected, the network description is generated (see Section 2.10). If the dataphone is disconnected, the output is printed on the teletype. Otherwise, the description is sent over the dataphone just as though it was typed into the teletype. The output specification light buttons are frozen in their current state to serve as labels on the displayed graph. After reception of the graph, the screen looks like this:
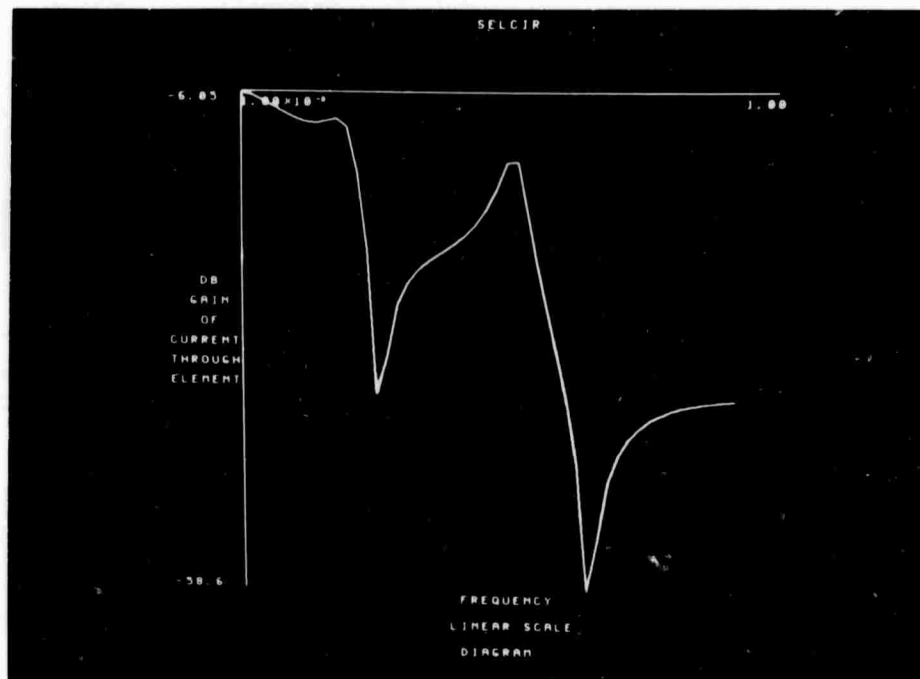
If the diagram description is merely being copied into a file, or if the dataphone is disconnected, no graph will, of course, appear. In any event, pointing to DIAGRAM will remove any graph from the screen and put the circuit diagram back. The output specification light buttons and switches are then unfrozen and operate as before. If no change is made to the circuit or output specification and GRAPH is pointed to, the previous graph reappears on the screen. If the circuit or output specification are changed, pointing to GRAPH requests new output from the 360. The old graph is destroyed, the diagram structure is retransmitted, and a new graph is sent back. Thus, the user can easily make changes in the circuit and see immediately how they affect its operation. To plot the same circuit parameter, he only needs to point to GRAPH. To plot another parameter, he only needs to flip the specification light switches which require changing and then point to GRAPH.

2.9 Abnormal Conditions during Output Specification

When the user points to GRAPH certain error conditions can arise. In this case, the appropriate comment is typed on the teleprinter and no output is generated.

NO DIAGRAM - there are no elements or connections on the screen

to be transmitted. The user is advised to draw some.

MANY FRAGMENTS - there is more than one disjoint set of

interconnected elements. Either delete the extra fragments

or hook them into the circuit somewhere.

UNCONNECTED TERMINALS - Some element(s) still has uncon-

    nected terminals. An indicator is placed on the screen

    around such terminals and will remain until the terminal is

    connected.

UNASSIGNED PARAMETERS - there are parameter dots on the

    screen which have not been assigned a numeric value. An

    indicator is placed on the screen around such parameter

    dots and will remain until the dot is assigned a value.

NO OUTPUT ELEMENT - no output element has been specified.

    Point to one.

IMPROPER RANGE - the low frequency specified is higher than

    the high frequency. Switch the values to obtain the correct

    range.

2.10 Network Description Format

Here follows the format of the network description generated by SELCIR and read in by the analysis program CIRAN.

Title Record: Expected by CIRAN but ignored otherwise

    Col 1-3   ADD

Element Records:

    Col 1   Type of element R - resistor
                                           C - capacitor
                                         L - inductor
                                       V - voltage source
                                       I - current course

    Col 2-4   Internally generated element number

Col 6	Control type (V or I) if a control source otherwise blank
Col 7-10	Name of controlling element if a controlled source otherwise blank
Col 12-13, 15-16	Internally generated node numbers of connections at each end of element. For passive elements these are ordered to indicate polarities:

horizontal orientation   vertical orientation
positive on left   positive on top

Col 18	V (voltage across) or I (current through) if output element otherwise blank
Col 20-28	Parameter value of element

Output Specification Records:

For Frequency Plot:

Col 1-4	FREQ
Col 6-8	LIN for linear scale LOG for logarithmic scale
Col 12	1 for plot of absolute gain 2 for plot of gain in db 3 for plot of relative phase angle
Col 20 . . .	Maximum and minimum frequencies separated by comma

For Time Plot:

    Col 1-4   TIME

    Col 6-8   STEP for step input
                                      IMP for impulse input

    Col 20 . . .   Maximum and minimum times separated by comma

For Poles and Zeros Plot:

    Col 1-3   P+Z

End of Diagram Record:

    Col 1-3   ADD

3. INTERNAL OPERATION

This section describes the programming techniques used to implement SELCIR. The four areas of interest are: the tasks which monitor I/O operation; the "thresholding" technique of light pen interpretation; the data structure used to store graphical and relational information; and the algorithm for adjusting the connection lines and element positions to produce neat looking diagrams.

3.1 Tasks

SELCIR runs under a multiprogramming system, the SEL executive system, so that several tasks may be scheduled to run simultaneously. A task is activated by an input from one of four I/O devices: teletype, dataphone, push button box, or light pen. The first three devices each have a single, unchanging service task. The task scheduled by a light pen hit, however, changes during different stages of the drawing and analysis process. These tasks are described below.

3.1.1 Keyboard Input Interpreter

This looks at characters typed into the teletype and carries out the operation indicated, as follows:

#	Copy rest of line to the dataphone, subject to the editing described in Section 2.9
CLEAR? OK	Removes the current diagram from the screen and releases its allocated storage. Clears out the local element name table.

30

ESCAPE ? OK   Removes all displays from the screen and releases
all the allocated storage.   Sends an end-of-file
record over the dataphone.   Terminates all other
scheduled tasks, releases all devices,and returns
to the system.

3.1.2 Dataphone Input Interpreter

This task reads all records transmitted to the PDP-9 from the
360 and processes them appropriately. ASCII records are simply
typed on the teletype while binary records are interpreted as graphs
sent by the analysis program. When reception of a new graph begins,
the old graph existing in the PDP-9 is destroyed. The new graph
display file is then constructed taking care that it always contains
valid display instructions. It can then by inserted into the active
display structure by other asynchronous tasks while construction of
the graph continues.

3.1.3 Push Button Interpreter

This task reads push buttons used to specify parameter values,
and constructs the numeric display file at the upper right-hand corner
of the screen. Pushing a button numbered 0 through 9 concatenates
that digit to the displayed number. Button 10 concatenates a decimal
point and changes the diagram light pen service task (see below).
Pushing button 11 removes the number from the screen and resets the
diagram light pen service task to its previous value.

3.1.4 Light Pen Service Tasks

Whenever the light pen detects light from a displayed picture, a service task is scheduled to respond to the light pen hit. The identity of the task to be scheduled depends on what is pointed to and on the current state of the program. In general, the light pen is completely disabled during execution of a service task and re-enabled at its termination. Since most service tasks execute almost instantaneously, there is a half-second time-out added before re-enabling the light pen. This allows the user to move the pen away from the screen before another light pen hit occurs.

3.1.4.1 Diagram Service Tasks

The circuit diagram has a task associated with it called the diagram service task. The identity of the diagram service task is changed whenever a different response to a light pen hit on the diagram is desired (e.g., by flipping TRANSLATE to TRANSLATE ON).

A typical diagram service task is PENINT, the service task initially set when the program starts. Pointing to the circuit schedules PENINT which examines the display structure to determine what part of the circuit was hit and branches accordingly.

Hit on connection line - Begin interpreting connection line

thresholds (See Section 3.2.2).

Hit on element terminal - Create the display structure for a

connection at the terminal and begin interpreting the

threshold pattern for the beginning of a connection
(See Section 3. 2. 3).

Hit on an element - Begin interpreting the Move/Delete/Rotate
threshold pattern. If the element is connected to other
elements, the rotation threshold is ignored (See Section 3. 2. 1).

Another diagram light pen service task is PENTRA. Pointing to the
TRANSLATE light switch changes the diagram service task to
PENTRA. Pointing to the diagram will then schedule PENTRA.
PENTRA moves the entire diagram, consisting of all connected and
unconnected elements, to follow the motion of the light pen. When
tracking is lost, it terminates.

3. 1. 4. 2 Light Buttons

There is a unique service task associated with each light button.
These tasks perform the operations indicated by the button and
described in the user's manual. A typical light button service task
is CONCRE which is scheduled whenever an element creation light
button is pointed to. The service task first uses information from an
internal element dictionary to build the display structure necessary
to display the proper element. Tracking is then started and the
element is continually repositioned to remain under the tracking
cross until tracking is terminated.

3.1.4.3 Light Switches

A light switch service task is actually a pair of light button service tasks, one for each state of the light switch. A task alters the light switch to display the opposite state and then performs the operation corresponding to the change of state of the switch. A typical example is CONTRA, scheduled when the TRANSLATE/TRANSLATE ON light switch is hit. If TRANSLATE is hit, it changes the display to TRANSLATE ON and sets the diagram service task to PENTRA. Light pen hits on the circuit diagram will then be serviced by PENTRA (see below). If TRANSLATE ON is hit, it changes the display to TRANSLATE and resets the diagram task to its original value. Light pen hits on the diagram will then be serviced as they were before TRANSLATE was turned on.

Note the effect of the half-second time-out before re-enabling the light pen. Since the operation of switching the diagram light pen service task is almost instantaneous, the time-out keeps the light pen from instantaneously seeing the new state and flipping back. With the time-out, the light pen will not recognize the new state for half a second, giving the user time to move the pen away. Then pointing at the switch re-schedules CONTRA which changes the light switch back. Simply holding the pen over a light switch causes it to alternate between states twice a second. At all times, the state which is displayed is the currently active state.

3.2 Thresholding

Thresholding is the technique whereby the user makes decisions by moving the tracking cross in a specified fashion. While a threshold pattern is being interpreted, the screen is marked off into a pattern of segments or regions. The region containing the tracking cross is noted, an appropriate action is taken and the pattern is re-checked. This checking is continually repeated until something explicitly stops it.

Sometimes several threshold patterns are superimposed, as for the move/delete/rotate operation. In this particular case, the three patterns are cyclicly examined, each one checking for a particular motion of the tracking cross. The first successful check terminates the process by branching to an appropriate service routine.

3.2.1 Move or Branch
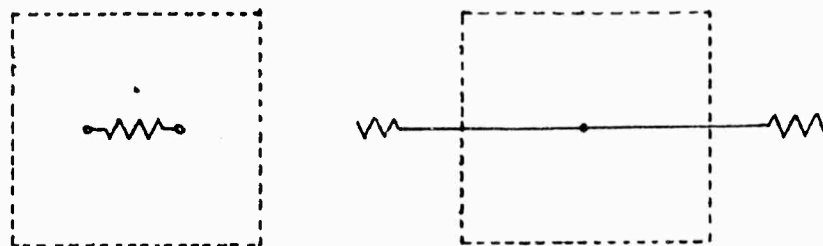
This pattern is used in the move/delete/rotate response to pointing at an element and in the branch/delete response to pointing at a connection. An imaginary box is initially drawn around the coordinates of the light pen hit. If the tracking cross is detected outside the box, the move (or branch) condition is satisfied and

threshold interpretation is terminated by branching to a routine that moves the element or creates the branch.

3.2.2 Delete
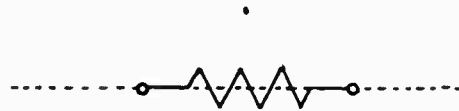
A counter is initialized to -5 at the beginning of the threshold interpretation loop. Each time the tracking cross crosses an imaginary horizontal line through the center of the element, the counter is incremented. If it overflows (before another thresholding condition is satisfied) threshold interpretation is terminated and the element or connection is deleted from the display structure.

Of course, some problems could arise if the light pen is not moved perfectly smoothly; pointing to an element with a shaky hand could accidentally delete it. For this reason, the line is actually a stripe. The light pen must move all the way across it to increment the stroke count and all the way back to increment the count again. Thus:

count=-4   count=-3   count=-2
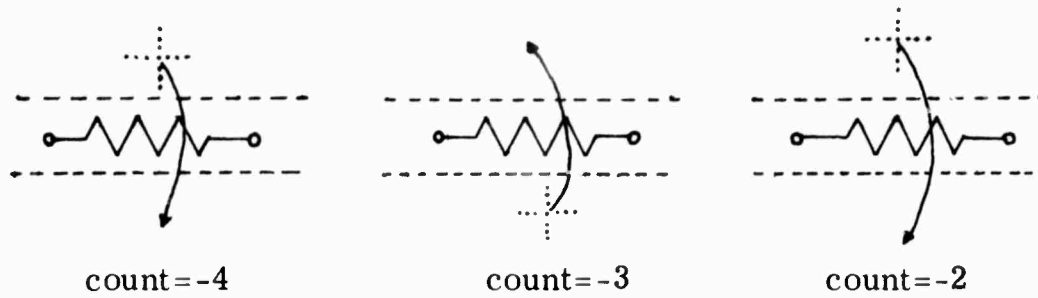
This type of hysteresis is added to all threshold patterns which could be bothered by sloppy light pen motion.

3.2.3 Rotate
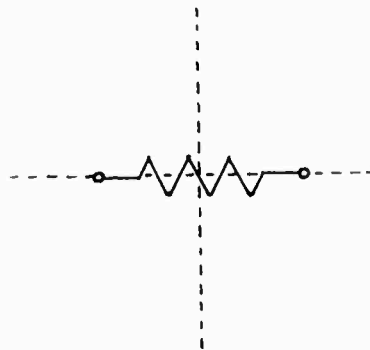
The rotate pattern is composed of horizontal and vertical lines drawn through the center of the element, dividing the screen into four quadrants. If the tracking cross is detected sequentially in the four quadrants in a circular path (while remaining inside the move box and while not stroking up and down more than four times) the rotate condition is satisfied. Threshold interpretation is terminated by branching to a routine which rotates the element 90°.

3.2.4 Line Drawing

Connection lines are constrained to be composed of only horizontal and vertical segments. While the line is being drawn, a threshold pattern is continually checked to determine when the user "goes around a corner" (necessitating the addition of a new line segment perpendicular to the old one) or when he backtracks (necessitating the removal of a line segment). The pattern is centered at the stationary end of the line segment currently being drawn
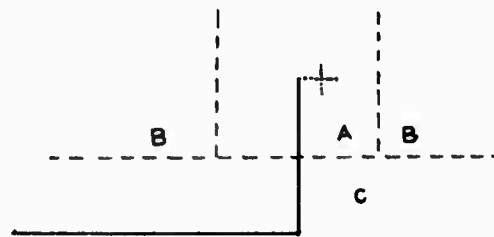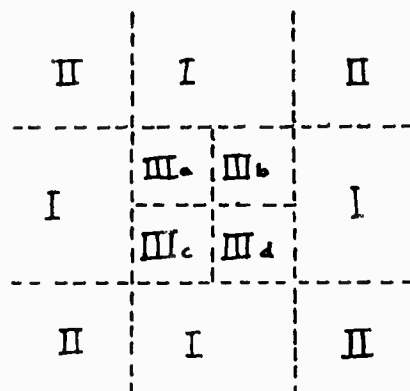

Pattern 1

If tracking is detected in region A, the Y coordinate of the vertical line is updated to match the tracking cross and the pattern is rechecked, If the user begins to draw horizontally, moving the cross into region B, a new line segment is added and the thresholding pattern is moved up to the top of the vertical line.
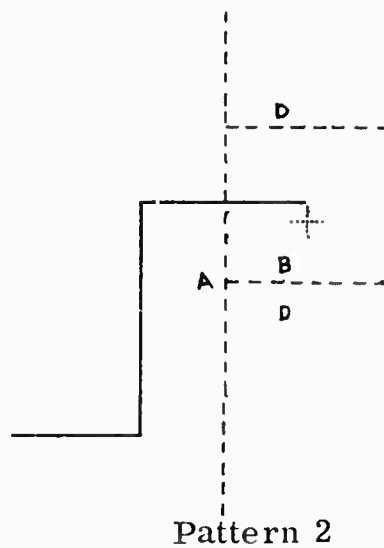
Pattern 2

Now, as long as tracking remains in region B of pattern 2, the endpoint of the horizontal segment is continually updated. If, however, the user backtracks into region A, the horizontal segment is deleted, the thresholding pattern returns to pattern 1 and the vertical segment once more follows the light pen. Now, if the user further backtracks into region C of pattern 1, the vertical segment is deleted and the previous horizontal one follows the light pen.

3.2.5 Line Initiation

Whenever the drawing of a connection line is initiated, care must be taken to prevent the user from drawing in a direction already occupied by another line or element. The following threshold pattern is placed at the initial coordinates of the connection line:
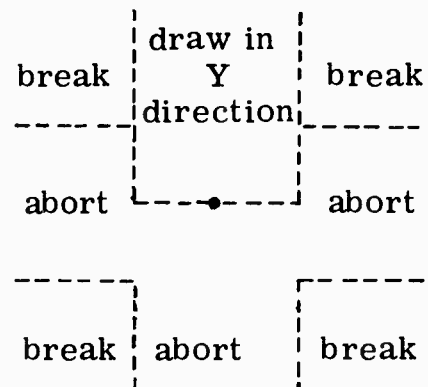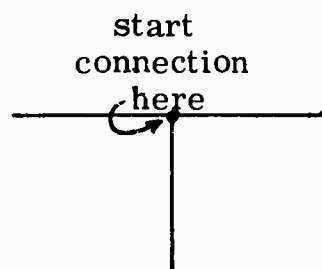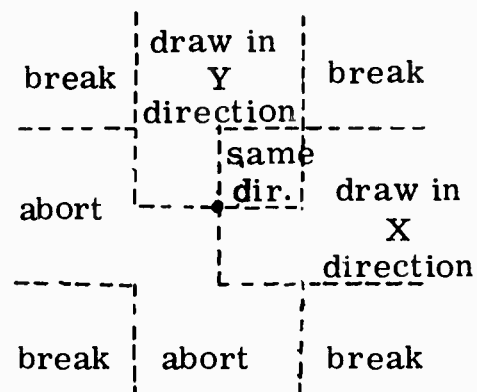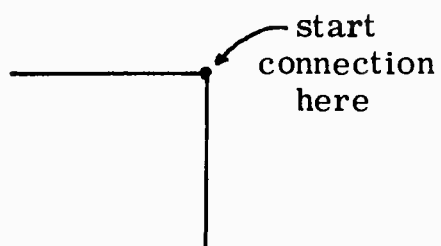
```
      |         |
  Ⅱ   |    I    |    Ⅱ
      |         |
-- ---|-- --.-- --|-- ---
      | Ⅲₐ |Ⅲᵦ |
  I   |----|-- -|    I
      | Ⅲᵤ |Ⅲₐ |
-- ---|-- --.-- --|-- ---
      |         |
  Ⅱ   |    I    |    Ⅱ
      |         |
```
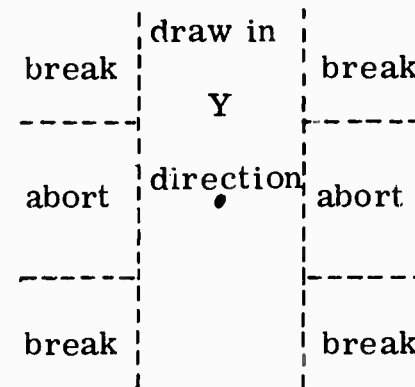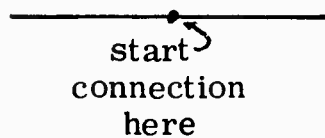
Detection of tracking in a region labeled I precipitates a different

action depending on whether that segment already contains a line.

If it is occupied, the line drawing operation is aborted. Otherwise, the

endpoint of the line segment is updated either horizontally or vertically

and the pattern is rechecked.

Detection in region II adds a new segment to the connection line,

perpendicular to the one one, and sets up a "line drawing" threshold

around it (See Section 3.2.4).

Regions IIIa ... IIId are at the four corners. Detection there

checks both associated directions for occupancy. If both directions

are occupied, the line drawing operation is aborted. If one direction

is free, the line is drawn in that direction. If both are free the line is

drawn in the same direction as the previous update.

Some examples of this threshold in practice are:

	draw in Y direction	
break		break
abort	same dir.	draw in X direction
break		break

	draw in Y direction	
break		break
abort		abort
break		break

	draw in Y direction	
break		break
abort	same dir.	draw in X direction
break	abort	break

	draw in Y direction	
break		break
abort		abort
break	abort	break

3.3 Data Structure

The basic data structure elements and manipulations used in
SELCIR are provided by the SEL executive system and are described
in [1]. Briefly, the system provides for a tree structured display
program. The main relational units, called "levels," contain
positional and display scaling data, etc. A display file which contains
a displayed picture is called a "leaf". Levels and leaves can be inserted
as sub-elements of other levels. A level or leaf is called an "attribute"
of the level it is inserted into. The display control then begins at a
system defined "Highest Active Level" and executes its attributes
and its attribute's attributes in the manner of a threaded list. Any-
thing to be displayed, then, must be inserted in the structure under
the highest active level. The diagrams below represent levels as dots
and leaves as boxes. The immediate attributes of the highest level
are:

The title, SELCIR, and the TRANSLATE light switch are always on the screen and are put immediately into the HAL. Light buttons and switches which are dynamically removed and inserted during execution of SELCIR are put in a separate button level. Graphs, etc., are placed in the scratch level, and the circuit is placed in the diagram level. The diagram level will contain the structure:

HAL

diagram level

fragment levels

symbol levels
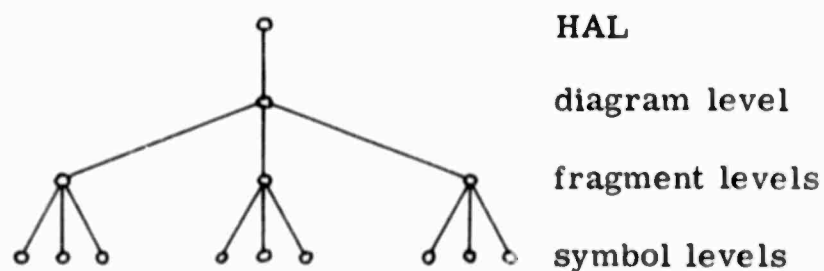
All elements and connections which are connected are placed into one fragment level, which is then placed in the diagram level. Then, when an element is moved, the displacement coordinates of its fragment level are altered, also moving everything else in the fragment. A basic network element (resistor, capacitor, connection, etc.) has the general structure:
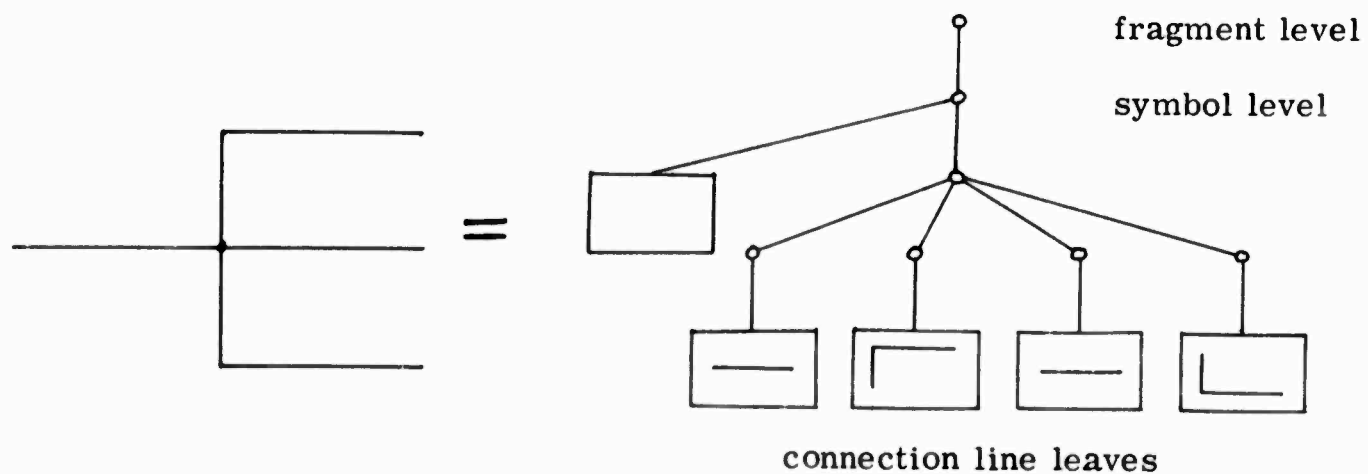
diagram level

fragment level

symbol level

body of
element

terminals   parameters
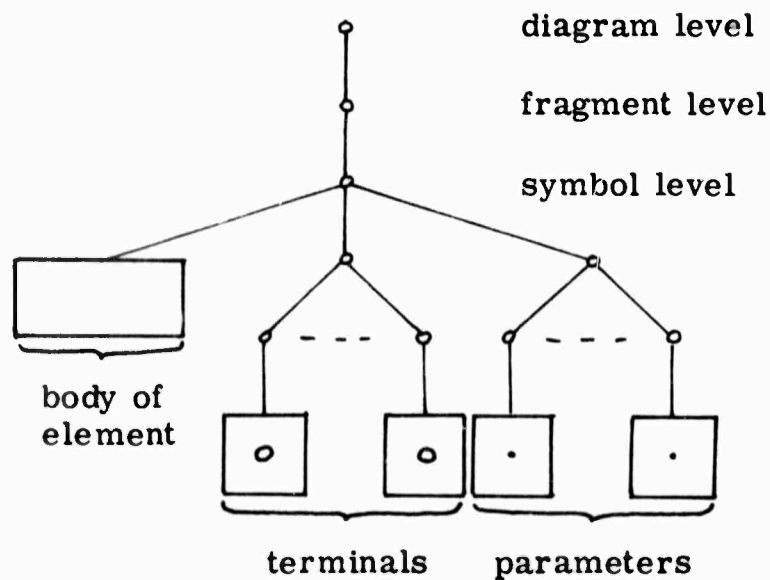
Each symbol level contains:

1st attribute: leaf to draw element body

2nd attribute: level containing connection terminals

3rd attribute: (if present) level containing parameter information.

The structure, then, allows for elements with any number of terminals
and parameters. Thus, expansion to the use of transistors or trans-
formers is straightforward. A typical element currently on the menu,
a resistor, has two terminals and one parameter. The structure is

fragment level

symbol level

body of resistor

terminal   terminal   parameter
dot
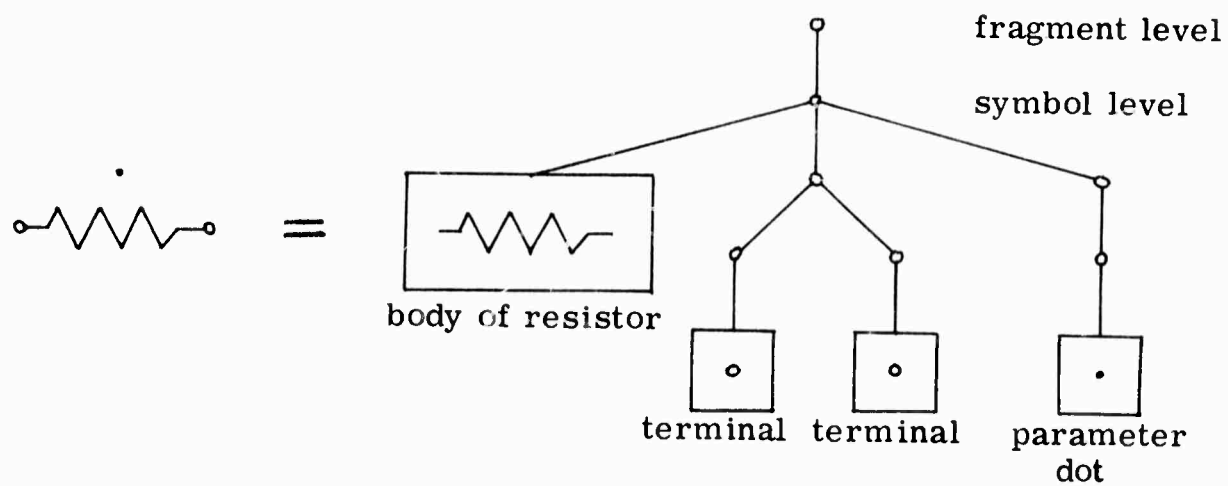
A connection may have many terminals but no parameter.

fragment level

symbol level

connection line leaves

An element with a connection line extending from one terminal has the structure:

diagram level
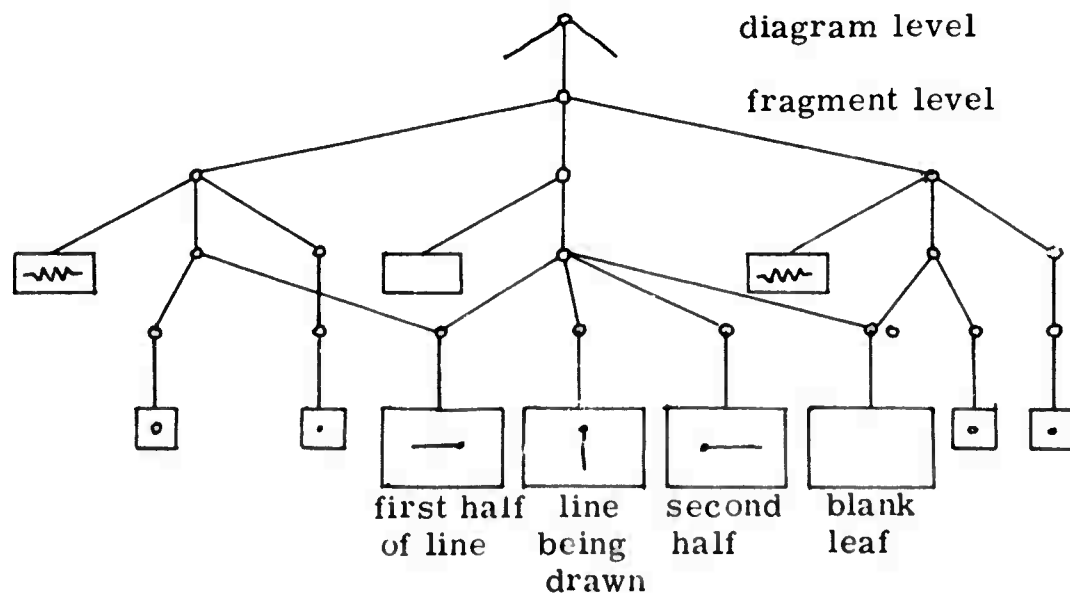
fragment level

element

connection
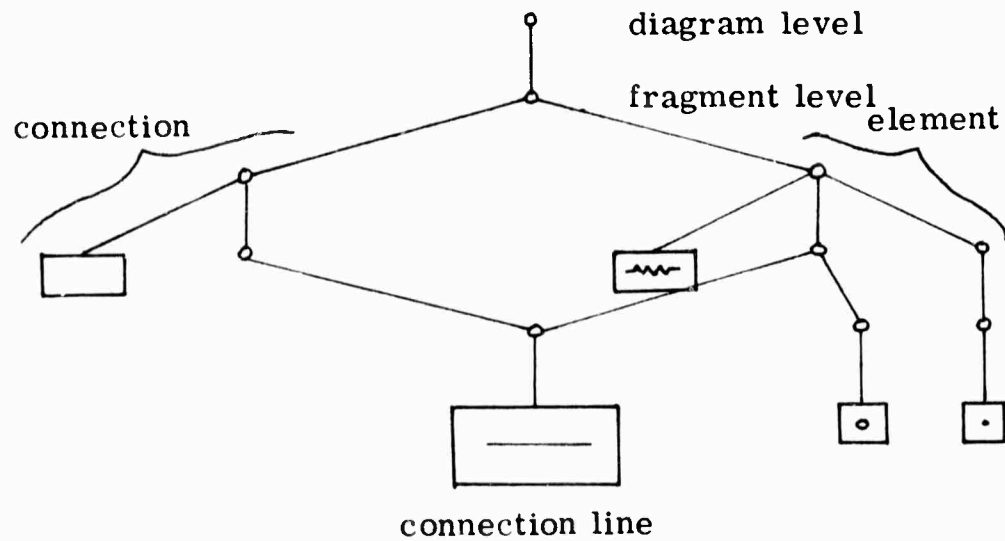
connection line

The element and the connection have one terminal in common. This is indicated in the structure by inserting the branch line leaf into both the element's and the connection's structure. The line leaf is the link in the data structure which indicates the relationship between the element and the connection. Knowing the core location of an element, the program can find which branch lines are connected to it through this link. Note that the structure of the element remains essentially unchanged except for the leaf inserted in its connection level. Then deleting the connection line entails a minimum of repair work to the structure of the element; only the leaf in the element's connection level needs to be switched. Occasionally, to preserve the structure of an element, it is necessary to use connection line leaves which display nothing. This is the case for the simple connection.
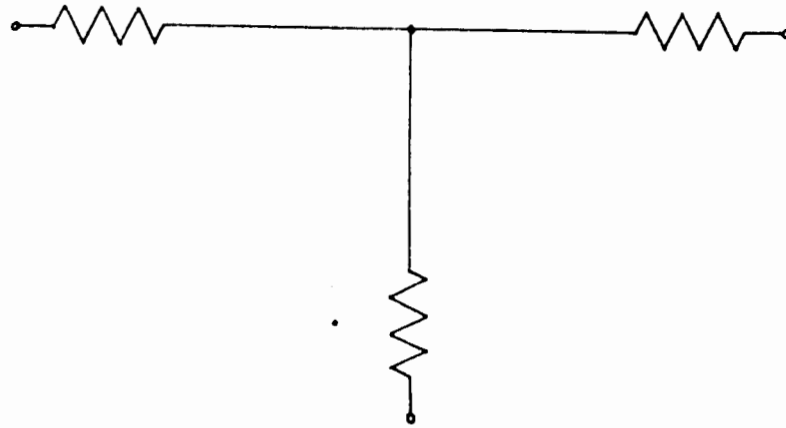
Which has the structure

diagram level

fragment level

connection
line leaf

blank leaf
(no displayed
information)
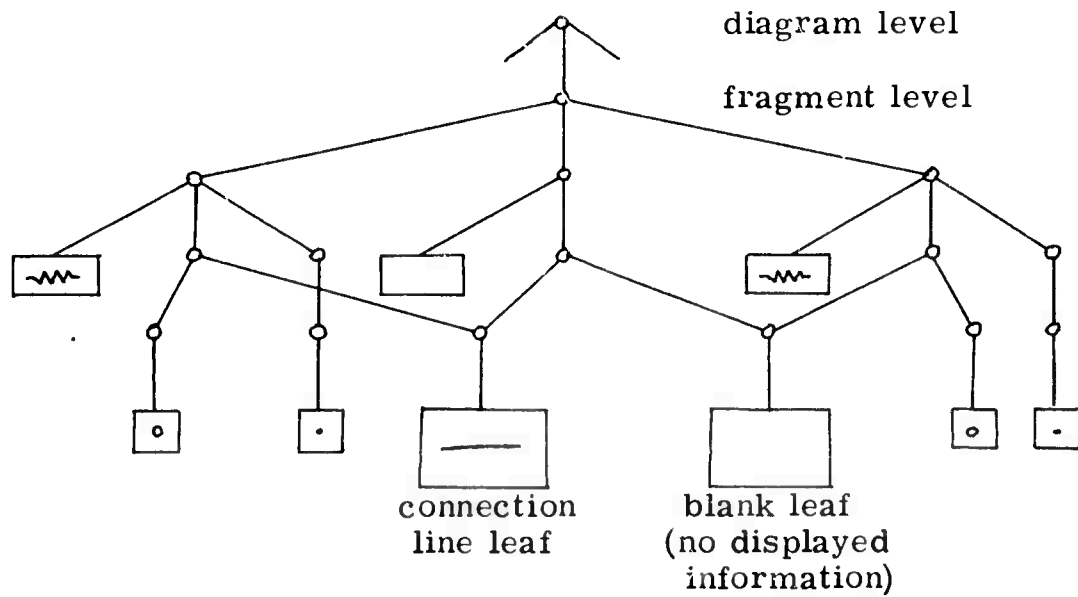
If a branch is then drawn from the connection line, the leaf is broken

into two leaves. The structure while the branch is being drawn is:

diagram level

fragment level

first half   line   second   blank
of line   being   half   leaf
             drawn

When the branch line is terminated at another element, we would
have the circuit:

with the structure

first half
of line
new branch
line leaf
second half
of line
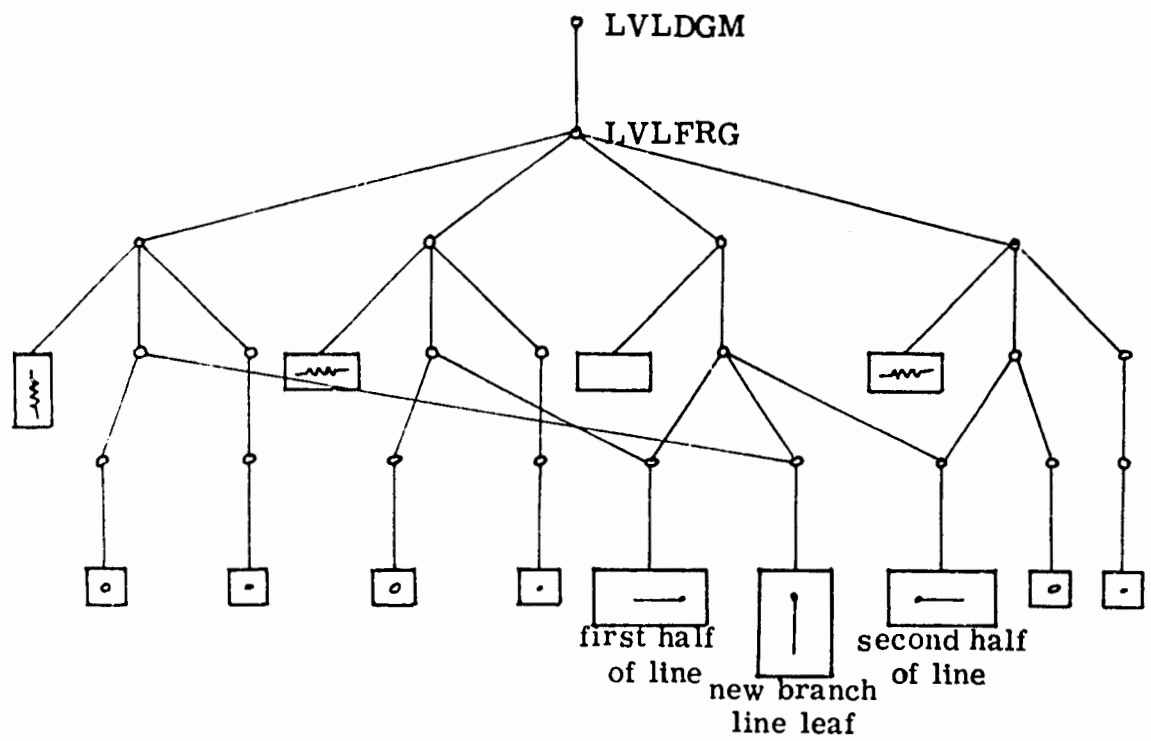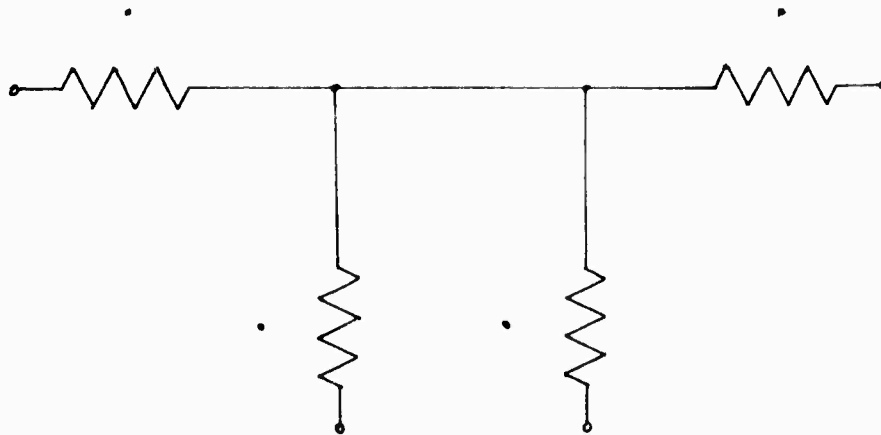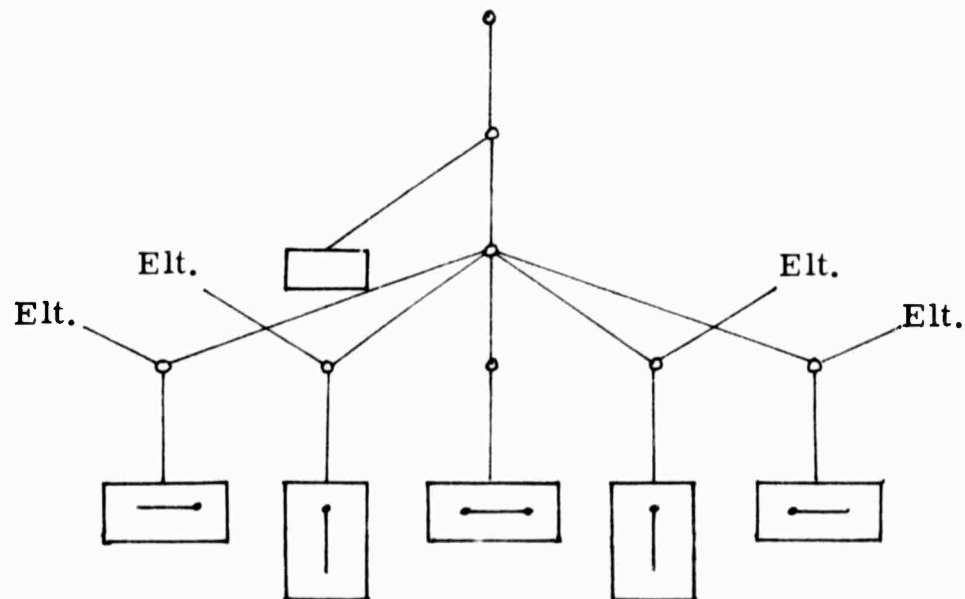
Note that the second half of the broken line leaf is merged with the empty leaf.

So far the diagrams shown have had an element at one end of each connection line. This is not always the case. For a diagram of form

the display structure of the connection would be

All line leaves are attributes of both the connection and an element, except for the leaf that has a junction at both ends.

3.4 Connection Initiation and Adjustment

A substantial amount of SELCIR is devoted to the drawing and adjustment of connection lines. The main effort is to constrain the user to draw "nice" looking connections by employing the following main criteria:

The user is not allowed to draw connection lines exactly coinciding with other lines.

A branch initiated near the endpoint of a line segment gravitates to the endpoint.

Positions of elements and shapes of connection lines are adjusted, if feasible, to straighten crooked lines.

When a connection line is begun, it is necessary to know the directions in which the line may be drawn without overlapping other lines. The set of already occupied directions is used to set up the connection initiation threshold pattern (See Section 3.2.5).

If the line is begun at an element terminal, there will be only one already used direction: the direction occupied by the element. When there is a light pen hit on a connection line, the line is first split into two separate leaves. If the hit was near one endpoint, the break is made at that end. Thus if the user points approximately at a corner, the junction will appear exactly at that corner instead of slightly offset from it. The breakpoint is then examined to determine the already occupied directions radiating from it. Note that a line

may be broken at the junction of several line segments. Here, all lines which abut this junction must be checked to see which direction they go in.

The connection line is thus initiated knowing one set of occupied directions. When the connection operation is successfully terminated, a set of occupied directions for the connection endpoint is found in an analogous manner. Since the light pen sees a fairly large area, the endpoint of the line being drawn rarely coincides exactly with the desired ending point of the connection. The adjustment routines must then adjust the end of the connection line by lengthening or shortening its line segments.

In the case where there is more than one line segment in the leaf, the adjustment is fairly simple. Consider the following example:
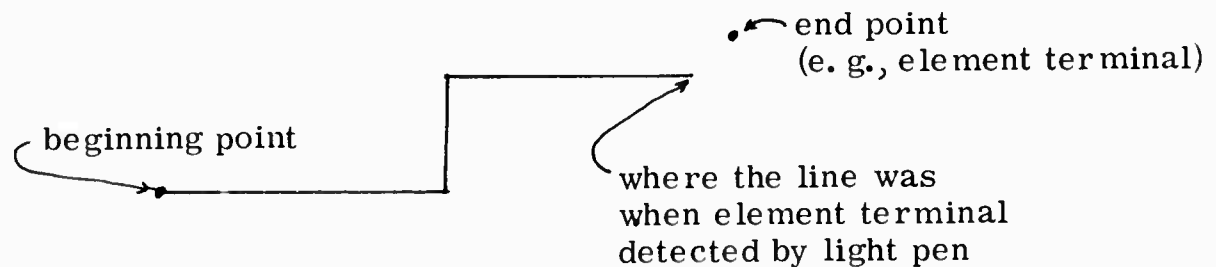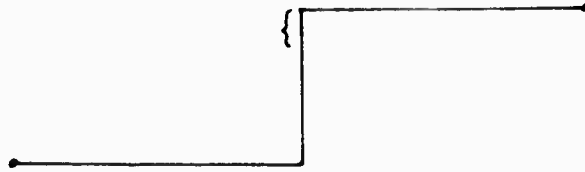

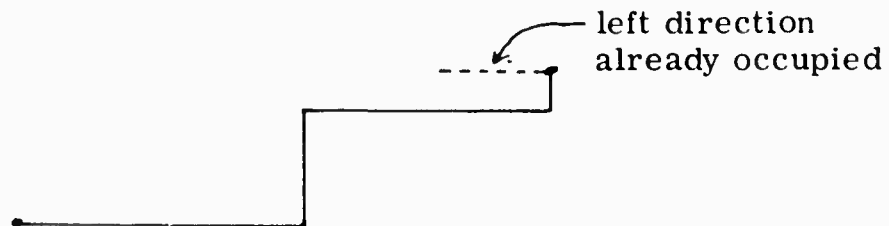First the line is extended the desired amount in the X direction.
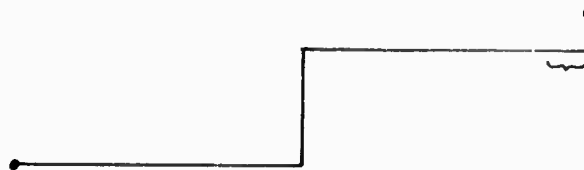
Then the occupied directions of the desired endpoint are examined.
If the left-hand direction is free, the Y vector is extended.

If the left-hand direction is occupied, another Y vector is added to
the line leaf.

— left direction
already occupied

Where there is only one drawn line in the leaf, the situation is
more interesting. First, a minimum length constraint is enforced.
If the line is shorter than the minimum length, it is extended, ensuring
that there is enough length for the light pen to see for later branches
or deletions. The line is extended only if the beginning and ending
point of the connection are in different fragments so that the elements
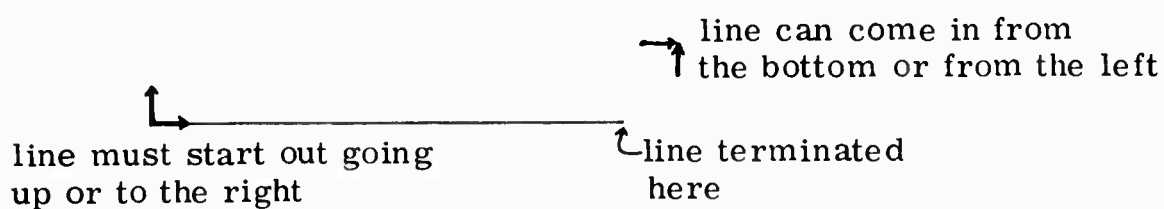can be moved apart. Thus

too close

is separated to

extended   moved
over
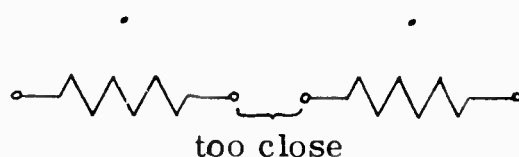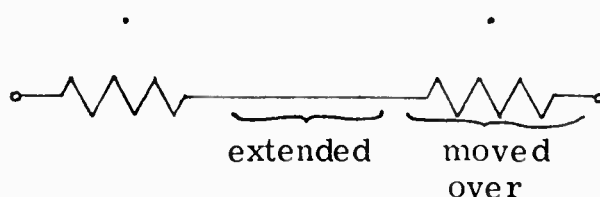
If the connected elements are already in the same fragment, the elements cannot easily be moved relative to each other and this constraint is ignored.

The direction perpendicular to the drawn line must then be adjusted. First, the coordinates of the desired starting and ending points are examined. At each end, the line can be adjusted to go in either of two directions to connect to the other end.

line can come in from
the bottom or from the left

line must start out going
up or to the right

line terminated
here

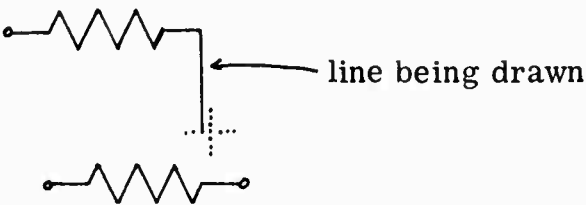Then various considerations are employed to choose the best looking starting and ending direction at each point. The already occupied directions are immediately eliminated. Then, if the starting point still has two choices, the direction of the drawn line is picked. If the ending point is an element terminal, the element orientation selects the ending direction. Otherwise the direction of the drawn line is used. Some examples of these choices are:

Diagram

Endpoint
position

Element direction
constraint

Diagram

Endpoint
position

Direction of drawn
line constraint

Diagram

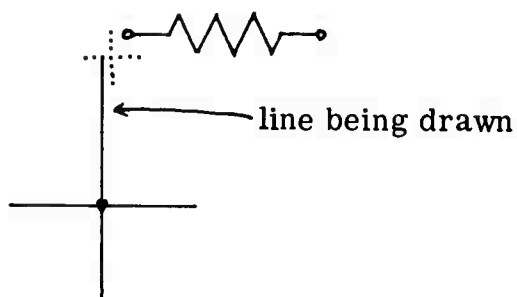line being drawn
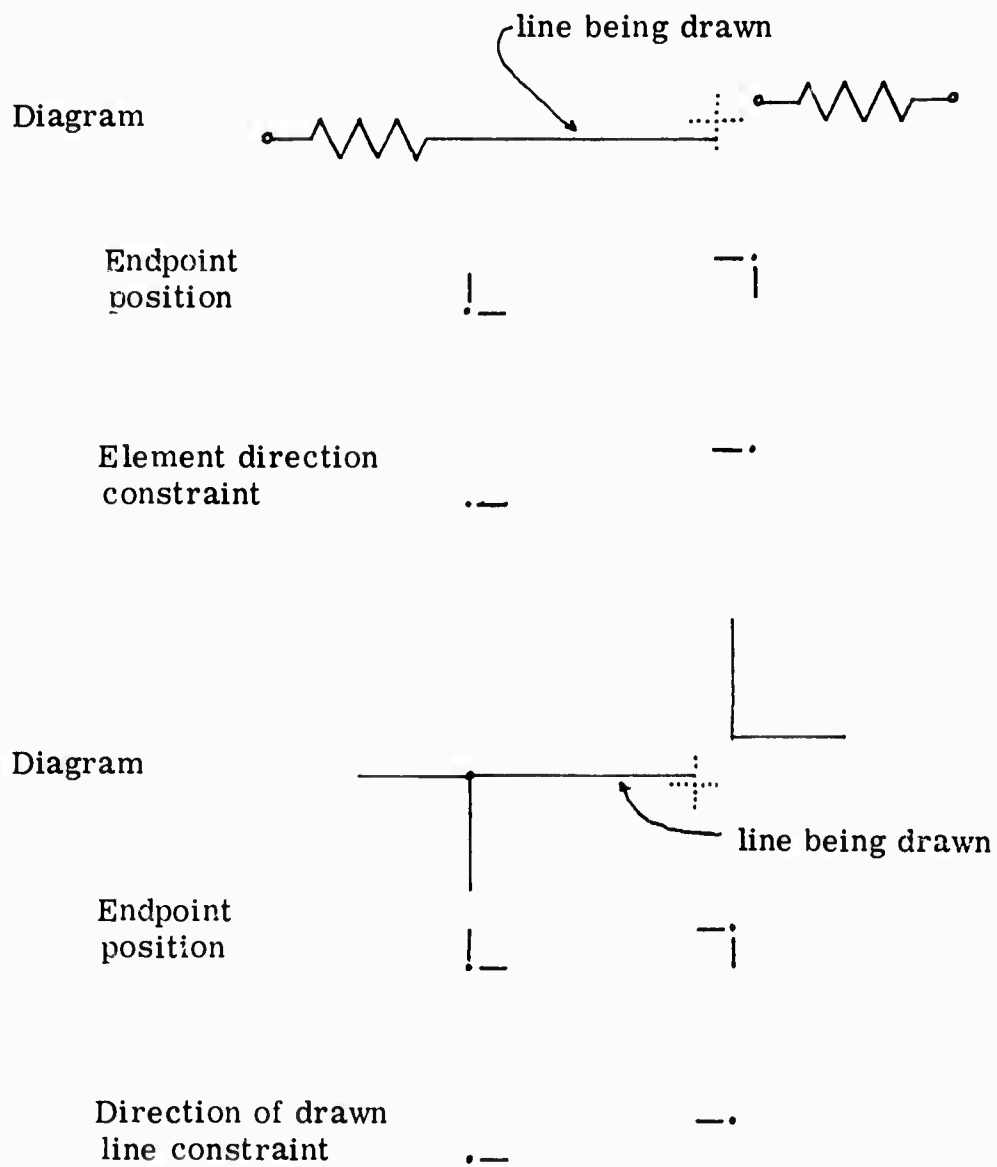
Endpoint
position

Already occupied
direction constraint

Direction of drawn
line constraint

line being drawn

Diagram

Endpoint
position

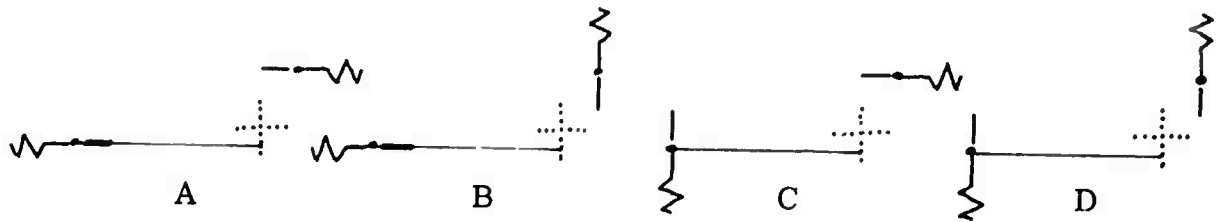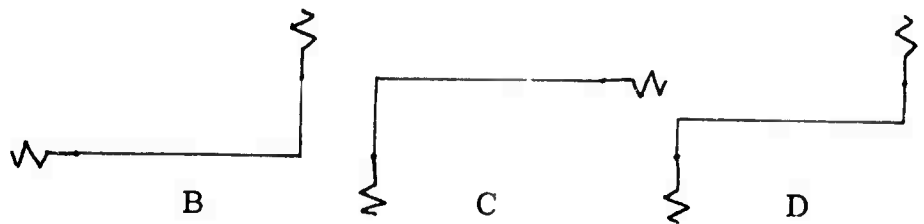Already occupied
direction constraint

Element direction
constraint

This, then, leaves four possible configurations (and their rotations and reflections):
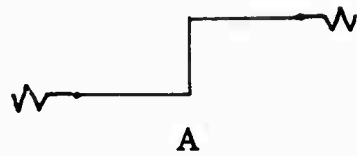

The adjustments for B, C, and D are

For configuration A it is still possible to avoid zig-zag lines. If the elements to be connected are in different fragments, the element position can be adjusted:

Otherwise, the line is adjusted:

A

4. POSSIBLE FUTURE EXPANSION

Work done so far has suggested the need of several additions and expansions to SELCIR. Some of these may be implemented by the time this report is printed. Others, unfortunately, may be unattainable dreams.

NEW ELEMENTS - The internal table structure of the circuit elements makes it easy to add or change elements. Thus elements such as diodes can be added when a nonlinear circuit analysis program is available in the 360.

MORE ELEMENTS - The current size limitation of about 13 elements can be raised to over 30 by using the more efficient data structure provided by a new version of the executive system.

SAVING NETWORKS ON REMOTE MASS STORAGE - A technique is being worked out for transmitting all graphical and relational information about a network to the 360 to be saved on a file there. The network could then be recalled and modified at a later date. An important feature of this must be the compactness of the information transmitted over the 201 line. It now appears that a screenfull of network can be transmitted in an acceptable two to three seconds.

NETWORK SEGMENTATION AND COMPOSITE ELEMENT DEFINITION - The limitation on the total size of the display network could be eliminated by allowing not all of the diagram to be displayed at once. Current plans include defining sub-networks as black boxes

which can be created, connected, etc. , just as normal elements.
The black boxes should also be able to be "opened" to examine their
insides.

REFERENCES

1. Jackson, James H., *An Executive System for a DEC 339 Computer Display Terminal*, Technical Report 15, Concomp Project, University of Michigan, Ann Arbor, December 196°.

2. *MTS Manual*, Second Edition, Computing Center, University of Michigan, Ann Arbor, December 1967.

61

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY *(Corporate author)* CONCOMP PROJECT UNIVERSITY OF MICHIGAN	2a. REPORT SECURITY CLASSIFICATION Unclassified	
	2b. GROUP	

3. REPORT TITLE

SELCIR: Systems Engineering Laboratory Circuit-Drawing Program

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Report

5. AUTHOR(S) *(First name, middle initial, last name)*

JAMES F. BLINN

6. REPORT DATE July 1970	7a. TOTAL NO. OF PAGES 61	7b. NO. OF REFS 2
8a. CONTRACT OR GRANT NO. DA-49-083 OSA 3050 b. PROJECT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report 24	
c. d.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* SEL Technical Report 47	

10. DISTRIBUTION STATEMENT

Qualified requesters may obtain copies of this report from DDC.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency

13. ABSTRACT

SELCIR, Systems Engineering Laboratory Circuit Drawer, enables the user to draw electronic circuit diagrams on a CRT display. All graphical manipulations are performed locally in a PDP-9, which, together with a 339 display scope, forms a remote terminal to an IBM 360/67. SELCIR scans the drawn network and transmits the topology of the network diagram to the central computer. The 360 then analyzes the network and sends back results which are displayed in the form of graphs. To enable the user to express himself naturally, the graphical manipulations emphasize the user of the light pen rather than push buttons. Also the user can stop at any point to correct mistakes or make changes. Finally, many constraints on the drawing process enable him to quickly make a neat looking diagram.

DD FORM 1473
NOV 65

Unclassified
Security Classification

63

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
interactive display graphics circuit analysis PDP-9						